DISTRIBUTING FAULT INDICATIONS AND
MAINTAINING AND USING A DATA STRUCTURE INDICATING FAULTS
TO ROUTE TRAFFIC IN A PACKET SWITCHING SYSTEM

5

FIELD OF THE INVENTION

This invention relates to maintaining, communicating, and reacting to faults in packet switching systems; and more particularly, the invention relates to distributing fault indications and maintaining and using a data structure indicating faults used to route traffic

10   through a packet switching system.

BACKGROUND OF THE INVENTION

The communications industry is rapidly changing to adjust to emerging technologies and ever increasing customer demand. This customer demand for new

15   applications and increased performance of existing applications is driving communications network and system providers to employ networks and systems having greater speed and capacity (e.g., greater bandwidth). In trying to achieve these goals, a common approach taken by many communications providers is to use packet switching technology.

As used herein, the term "packet" refers to packets of all types, including, but not

20   limited to, fixed length cells and variable length packets. Moreover, these packets may contain one or more types of information, including, but not limited to, voice, data, video, and audio information. Furthermore, the term "system" is used generically herein to describe any number of components, packet switch elements, packet switches, networks, computer and/or communication devices or mechanisms, or combinations thereof.

25   Consumers and designers of these systems typically desire high reliability and increased performance at a reasonable price. A commonly used technique for helping to achieve this goal is for systems to provide multiple paths between a source and a destination. Packets of information are then dynamically routed and distributed among these multiple paths. It is typically more cost-effective to provide multiple slower rate links

1

or switching paths, than to provide a single higher rate path. Such designs also achieve other desired performance characteristics.

It is important that packet switching systems are fault tolerant and appropriately detect and react to faults. Prior approaches to adapt to and overcome faults within a
5 packet switch typically rely on full redundancy in order to enable fault-masking of failures. For example, such a system might include one or more extra interconnection networks that could become active and replace another interconnection network which has a detected failure. Such approaches are typically costly. Needed are new apparatus and methods for reacting to faults within a packet switching system.

10

SUMMARY OF THE INVENTION

Apparatus and methods are disclosed for propagating and reacting to detected faults in a packet switching system. In one embodiment, a packet switching system
15 includes multiple input components of the packet switching system sending multiple packets to a multiple output components over multiple interconnection networks. After the packet switching system recognizes an error within the packet switching system, the packet switching system notifies the multiple input components of the error.

20 ## BRIEF DESCRIPTION OF THE DRAWINGS

The appended claims set forth the features of the invention with particularity. The invention, together with its advantages, may be best understood from the following detailed description taken in conjunction with the accompanying drawings of which:

FIGs. 1A-C are block diagrams of a few of many possible embodiments of a
25 switching system;

FIGs. 2A-C are block diagrams illustrating exemplary embodiments of a packet switching system component, such as, for example, a line card and/or input/output interface;

FIGs. 3A-C are block diagrams of exemplary switching fabric components;

FIG. 4 is a block diagram illustrating the operation of a broadcast mechanism within one of the interconnection networks for broadcasting fault information in a packet switching system;

FIGs. 5A-B illustrate exemplary packet formats used in propagating fault information;

FIGs. 6A-B illustrate exemplary data structures used to maintain fault information in a component of a packet switching system;

FIGs. 7A-C are embodiments that propagate and react to detected fault conditions; and

FIG. 8 illustrates manipulating data structures in determining a path through the packet switching system for a particular packet.

DETAILED DESCRIPTION

Methods and apparatus are disclosed for accumulating, distributing and reacting to detected faults and other error conditions in a packet switching system. Such methods and apparatus are not limited to a single packet switching environment. Rather, the architecture and functionality taught herein are extensible to an unlimited number of packet switching environments and embodiments in keeping with the scope and spirit of the invention. Embodiments described herein include various elements and limitations, with no one element or limitation contemplated as being a critical element or limitation. Each of the claims individually recite an aspect of the teachings disclosed herein in its entirety. Moreover, some embodiments described may include, *inter alia*, systems, integrated circuit chips, methods, and computer-readable medium containing instructions. The embodiments described hereinafter embody various aspects and configurations within the scope and spirit of the invention.

In one embodiment, a packet switching system detects faults and propagates indications of these faults to the input interfaces of a packet switch. The packet switch

may select a route for a particular packet to avoid these faults. In this manner, packet loss may be decreased upon a failure condition, and with only a short interruption delay being limited by the time to detect a failure and to send updates to the input sources.

Faults are identified by various components of the packet switching system and
5   relayed to one or more switching components to generate a broadcast packet destined for all input ports (e.g., to each I/O interface in a packet switch having folded input and output interfaces). Other embodiments generate one or more multicast or unicast packets. An I/O interface may be used to maintain one or more data structures indicating the state of various portions of the packet switching system. In one embodiment, an output
10   availability table is maintained indicating a path over which a particular destination may be reached, as well as a link availability vector indicating which output links of the input interface may be currently used. Using these as masks against possible routes in a fully functional system, the packet switching component (e.g., I/O interface) can identify which routes are currently available for reaching the destination of the received packet. These
15   routes can then be selected from among those using numerous deterministic and non-deterministic methods.

FIGs. 1A-3C and their discussion herein are intended to provide a description of various exemplary packet switching systems. FIGs. 1A-C illustrate the basic topology of different exemplary packet switching systems. FIG. 1A illustrates an exemplary packet
20   switch 100 having multiple inputs and outputs and a single interconnection network 110. FIG. 1B illustrates an exemplary packet switch 140 having multiple interconnection networks 141 and folded input and output interfaces 149. FIG. 1C illustrates an exemplary folded packet switch 160 having multiple interconnection networks 161 and folded input and output interfaces 169. Embodiments of each of these packet switches
25   100, 140 and 160 receive, generate, accumulate, distribute, and react to detected faults in the manners disclosed herein. Of course, the invention is not limited to these illustrated operating environments and embodiments, and the packet switching systems may have more or less elements.

4

FIG. 1A illustrates an exemplary embodiment of a packet switch 100. Packet switch 100 comprises multiple input interfaces 105, interconnection network 110, and output interfaces 125. Input interfaces 105 and output interfaces 125 are both coupled over multiple links to interconnection network 110. Line cards 101 and 131 are coupled

5   to input interfaces 105 and output interfaces 125. In certain embodiments including other packet switching topologies, line cards or their functionality may be included in the packet switch itself, or as part of the packet switching system.

In one embodiment, interconnection network 110 comprises multiple switch elements SE-1 112, SE-2 115, and SE-3 118 that are interconnected by multiple links.

10   Line cards 101 and 131 may connect to other systems (not shown) to provide data items (e.g., packets) to be routed by packet switch 100. Fault indications may be generated, consumed, or processed at one or more of the line cards 101, 131, input interfaces 105, switch elements SE-1 112, SE-2 115, and SE-3 118, output interfaces 125, and/or other locations within packet switch 100 or the packet switching system.

15   FIG. 1B illustrates another exemplary operating environment and embodiment of a packet switch 140. Packet switch 140 comprises multiple folded input and output interfaces 149 interconnected over multiple links to interconnection networks 141, which are interconnected over multiple links returning to input and output interfaces 149. In one embodiment, interconnection networks 141 comprise multiple switch elements SE-1 142,

20   SE-2 145, and SE-3 148 also interconnected by multiple links. Interfaces 149 may connect via bi-directional links to line cards 139 that connect with other systems (not shown) to provide data items (e.g., packets) to be routed by packet switch 140. Fault indications may be generated, consumed, or processed at one or more of the line cards 139, input and output interfaces 149, switch elements SE-1 142, SE-2 145, and SE-3

25   148, and/or other locations within packet switch 140 or the packet switching system.

FIG. 1C illustrates another exemplary operating environment and embodiment of a packet switch 160. Packet switch 160 has a folded network topology. Packet switch 160 comprises multiple folded input and output interfaces 169 interconnected over multiple

links to interconnection networks 161, which are interconnected over multiple links returning to interfaces 169. In one embodiment, interconnection networks 161 comprise multiple switch elements SE-1 & SE-3 162 and SE-2 164 also interconnected by multiple links. Interfaces 169 may connect via bi-directional links to line cards 159 which connect

5 via ports 158 to other systems (not shown) to provide data items to be routed by packet switch 160. Fault indications may be generated, consumed, or processed at one or more of the line cards 159, input and output interfaces 169, switch elements SE-1 & SE-3 162 and SE-2 164, and/or other locations within packet switch 160 or the packet switching system.

FIGs. 2A-C illustrate three of numerous possible embodiments of a line card, input

10 interface, output interface, and/or input/output interface. For illustrative purposes, only single transmitters and receivers may be shown. It should be clear to one skilled in the art that multiple transmitters and receivers may be used to communicate with multiple sources and destinations (e.g., line cards, switch fabrics, etc.).

FIG. 2A illustrates one component 220 comprising a processor 221, memory 222,

15 storage devices 223, and one or more external interface(s) 224, and one or more packet switch interface(s) 225, and one or more internal communications mechanisms 229 (shown as a bus for illustrative purposes). External interface(s) 224 transfer external signals with one or more communications devices or networks (e.g., one or more networks, including, but not limited to the Internet, intranets, private or public telephone,

20 cellular, wireless, satellite, cable, local area, metropolitan area and/or wide area networks). Memory 222 is one type of computer-readable medium, and typically comprises random access memory (RAM), read only memory (ROM), integrated circuits, and/or other memory components. Memory 222 typically stores computer-executable instructions to be executed by processor 221 and/or data which is manipulated by processor 221 for

25 implementing functionality in accordance with certain embodiments of the invention. Storage devices 223 are another type of computer-readable medium, and typically comprise disk drives, diskettes, networked services, tape drives, and other storage devices. Storage devices 223 typically store computer-executable instructions to be

executed by processor 221 and/or data which is manipulated by processor 221 for implementing functionality an apparatus disclosed herein. Component 220 generates, consumes, processes and reacts to fault indications.

5 As used herein, computer-readable medium is not limited to memory and storage devices; rather computer-readable medium is an extensible term including other storage and signaling mechanisms including interfaces and devices such as network interface cards and buffers therein, as well as any communications devices and signals received and transmitted, and other current and evolving technologies that a computerized system can interpret, receive, and/or transmit.

10 FIG. 2B illustrates one component 240 having a single element providing the functionality of a line card and an input/output interface, for example that of line card 159 and input/output interface 169 (FIG. 1C). Figures 2B-C will be described in relation to FIG. 1C for illustrative purposes; however, these embodiments could be used with other packet switch topologies and other implementations and embodiments. Component 240

15 comprises control logic 241 implementing functionality disclosed herein. In one embodiment, control logic 241 includes memory for storage of data and instructions. Control logic 241 is connected to other elements of component 240 via one or more internal communications mechanisms 249 (shown as a bus for illustrative purposes). External interface receiver 250 receives external signals, converts these signals using

20 demultiplexor 251 into multiple streams of packets which are temporarily stored in incoming packet buffer 252. At the appropriate time, a packet is sent to the appropriate switch element SE-1 & SE-3 162 via transmitter to switch elements 253. Packets are received from switch elements SE-1 & SE-3 162 at the receiver from switch elements 263 and placed in the outgoing packet buffer 262. Multiplexor 261 extracts the packets and

25 creates a multiplexed signal that is transmitted via external interface transmitter 260. Additionally, control logic 241 receives, generates, processes and reacts to fault indications as described hereinafter.

FIG. 2C illustrates one embodiment of a line card 270 and a switch interface 290, which could correspond to line card 159 and input/output interfaces 169 illustrated in FIG. 2C.

Line card 270 illustrated in FIG. 2C includes control logic 271 to control

5   operations of the input/output interface 270. Control logic 271 is connected to other components of line card 270 via one or more internal communications mechanisms 279 (shown as a bus for illustrative purposes). In one embodiment, control logic 271 includes memory for storing instructions and data. Line card 270 also includes optional additional memory 272 and storage devices 273. External interface receiver 274 receives external

10   signals 201 (FIG. 2), separates the signals into channels using demultiplexor 275 into multiple streams of packets which are temporarily stored in incoming packet buffer 276. At the appropriate time, a packet is sent to switch interface 290 via transmitter to switch interface 277. Packets are received from switch interface 290 at the receiver from switch interface 287 and placed in the outgoing packet buffer 286. Multiplexor 285 extracts the

15   packets and creates a multiplexed signal which is transmitted via external interface transmitter 284. In one embodiment, control logic 271, referencing a data structure within control logic 271 or memory 272, stores fault indications. Line card 270 may receive, generate, process and react to fault indications as described hereinafter. In certain embodiments, fault conditions may be hidden from a line card by other components which

20   react to the fault indications.

The embodiment of input/output interface 290 illustrated in FIG. 2C includes control logic 291 implementing functionality in accordance with certain embodiments of the invention. Control logic 291 is connected to other components of switch interface 290 via one or more internal communications mechanisms 289 (shown as a bus for illustrative

25   purposes). In one embodiment, control logic 291 includes memory for storing instructions and data. Switch interface 290 also includes optional additional memory 292 and storage devices 293. Line card receiver 294 receives packets from line card 270 temporarily stores the packets in incoming packet buffer 295. At the appropriate time, a packet is sent

to an appropriate switch element SE-1 & SE-3 162 via transmitter to switch elements 296. Packets are received from switch elements SE-1 & SE-3 162 at the receiver from switch elements 299 and placed in the outgoing packet buffer 298. Line card interface transmitter 297 then forwards these packets to line card 270. In one embodiment, control

5   logic 291, referencing a data structure within control logic 291 or memory 292, stores fault indications which could be received from a line card, packet switch, or internally generated. Input/output interface 290 receives, generates, processes and reacts to fault indications as described hereinafter.

       FIGs. 3A-C illustrate exemplary embodiments of switching elements and/or their

10   components. FIG. 3A is a block diagram of one embodiment of a first stage switching element, SE-1 300. FIG. 3B is a block diagram of one embodiment of a second stage switching element SE-2 330. FIG. 3C is a block diagram of one embodiment of a third stage switching element SE-3 360. As would be understood by one skilled in the art, the invention is not limited to these or any other embodiment described herein.

15   FIG. 3A illustrates an embodiment of SE-1 300 comprising control logic and/or processor 311 (hereinafter "control logic"), memory 312, storage devices 310, I/O interfaces 305, output queues 320, SE-2 interfaces 325, and one or more internal communications mechanisms 319 (shown as a bus for illustrative purposes). In certain embodiments, control logic 311 comprises custom control circuitry for controlling the

20   operation of SE-1 300 and no storage device 310 is used. Memory 312 is one type of computer-readable medium, and typically comprises random access memory (RAM), read only memory (ROM), integrated circuits, and/or other memory components. Memory 312 typically stores computer-executable instructions to be executed by control logic 311 and/or data which is manipulated by control logic 311 for implementing functionality in

25   accordance with certain embodiments of the invention. Storage devices 310 are another type of computer-readable medium, and may comprise, for example, disk drives, diskettes, networked services, tape drives, and other storage devices. Storage devices 310 typically

store computer-executable instructions to be executed by control logic 311 and/or data which is manipulated by control logic 311 for implementing functionality disclosed herein.

SE-1 300 generates, consumes, processes and reacts to fault indications as further described in detail hereinafter. Briefly first, each SE-1 300 receives packets 301 and

5   exchanges control messages 302 over one or more links with one or more input interfaces (not shown) such as input/output interface 290 (FIG. 2C) via I/O interfaces 305. Additionally, each SE-1 300 sends packets 328 and exchanges control messages 329 over one or more links with one or more SE-2 elements (not shown) such as SE-2 330 (FIG. 3B) via SE-2 interfaces 325. Control logic 311 detects faults, generates control

10   packets containing indications of the detected faults, and updates its fault data structure stored in memory 312. SE-1 300 may distribute fault indications to other packet switching components by sending control packets to other components as well as including fault indications in reserved fields of other control messages (e.g., acknowledgment or clear-to-send control messages) being sent. Outgoing packets and

15   control messages are placed in output queues 320. In one embodiment, there is an output queue 320 for each destination, or for each class of service for each destination.

FIG. 3B illustrates an embodiment of SE-2 330 comprising control logic and/or processor 341 (hereinafter "control logic"), memory 342, storage devices 340, SE-1 interfaces 335, output queues 350, SE-3 interfaces 355, and one or more internal

20   communications mechanisms 349 (shown as a bus for illustrative purposes). In certain embodiments, control logic 341 comprises custom control circuitry for controlling the operation of SE-2 330 and no storage device 340 is used. Memory 342 is one type of computer-readable medium, and typically comprises random access memory (RAM), read only memory (ROM), integrated circuits, and/or other memory components. Memory 342

25   typically stores computer-executable instructions to be executed by control logic 341 and/or data which is manipulated by control logic 341 for implementing functionality described herein. Storage devices 340 are another type of computer-readable medium, and may comprise, for example, disk drives, diskettes, networked services, tape drives,

and other storage devices. Storage devices 340 typically store computer-executable instructions to be executed by control logic 341 and/or data which is manipulated by control logic 341 for implementing functionality described herein.

SE-2 330 generates, consumes, processes and reacts to fault indications as further described in detail hereinafter. Briefly first, each SE-2 330 receives packets 331 and exchanges control messages 332 over one or more links with one or more SE-1 elements (not shown) such as SE-1 300 (FIG. 3A) via SE-1 interfaces 335. Additionally, each SE-2 330 sends packets 358 and exchanges control messages 359 over one or more links with one or more SE-3 elements (not shown) such as SE-3 360 (FIG. 3C) via SE-3 interfaces 355. In one embodiment using a folded topology, the links between (a) SE-2 330 and SE-1 300 and (b) SE-2 330 and SE-3 360 are the same links. Control logic 341 receives control packets containing fault indications, and updates its fault data structure stored in memory 342. Additionally, fault indications are broadcast through the packet switch or packet switching system, such as to all the I/O interfaces. Depending on the capabilities of the packet switching system, either a packet broadcast or multicast function could be used to efficiently distribute the fault indications; otherwise multiple packets are sent. Additionally, it is possible that fault indications are collected and sent on a periodic or on an event basis. However, in one embodiment fault indications are distributed immediately.

SE-2 330 may distribute fault indications to other packet switching components by sending control packets as well as including fault indications in reserved fields of other control messages (e.g., acknowledgment or clear-to-send control messages) being sent. Outgoing packets and control messages are placed in output queues 350. In one embodiment, there is an output queue 350 for each destination, or for each class of service for each destination.

FIG. 3C illustrates an embodiment of SE-3 360 comprising control logic and/or processor 371 (hereinafter "control logic"), memory 372, storage devices 370, SE-2 interfaces 365, output queues 380, I/O interfaces 385, and one or more internal

communications mechanisms 379 (shown as a bus for illustrative purposes). In certain

embodiments, control logic 371 comprises custom control circuitry for controlling the

operation of SE-3 360 and no storage device 370 is used. Memory 372 is one type of

computer-readable medium, and typically comprises random access memory (RAM), read

5   only memory (ROM), integrated circuits, and/or other memory components. Memory 372

typically stores computer-executable instructions to be executed by control logic 371

and/or data which is manipulated by control logic 371 for implementing functionality

described herein. Storage devices 370 are another type of computer-readable medium,

and may comprise, for example, disk drives, diskettes, networked services, tape drives,

10   and other storage devices. Storage devices 370 typically store computer-executable

instructions to be executed by control logic 371 and/or data which is manipulated by

control logic 371 for implementing functionality described herein.

SE-3 360 generates, consumes, processes and reacts to fault indications as further

described in detail hereinafter. Briefly first, each SE-3 360 receives packets 361 and

15   exchanges control messages 362 over one or more links with one or more SE-2 elements

(not shown) such as SE-2 330 (FIG. 3B) via SE-2 interfaces 365. Additionally, SE-3 360

sends packets 388 and exchanges control messages 389 over one or more links with one

or more output interface elements (not shown) such as Input/Output interface 390 (FIG.

2C) via I/O interfaces 385. Control logic 371 receives control packets containing fault

20   indications, and updates its fault data structure stored in memory 372. SE-3 360 may

distribute fault indications to other packet switching components by sending control

packets as well as including fault indications in reserved fields of other control messages

(e.g., acknowledgment or clear-to-send control messages) being sent. Outgoing packets

and control messages are placed in output queues 380. In one embodiment, there is an

25   output queue 380 for each destination, or for each class of service for each destination.

FIG. 4 illustrates a logical diagram of the operation of an embodiment for

distributing fault indications to I/O interfaces 410. In certain embodiments of packet

switching systems, it is important to broadcast the status of detected faults to all I/O

interfaces 410. FIG. 4 illustrates the operation of the collection and broadcast of an indication of an identified fault using a packet switching system having multiple line cards 401, each connected to an I/O interface 410. Note, the topology illustrated in FIG. 4 is that of a folded packet switch, and that each line card 401 and I/O interface 410 are shown

5   both on the left and right side of FIG. 4 for simplicity of illustration. Also, switch elements SE-1 411 and SE-3 413 are illustrated separately; however in certain embodiments such as that illustrated in FIG. 1C, these are embodied in the same component.

For illustrative purposes, the example shown in FIG. 4 assumes a fault is detected

10   in SE-1 component 411B. The teachings of FIG. 4 and its discussion can be directly applied to other components. Upon detection of a fault, SE-1 component 411B generates a control packet containing the fault indication. An example of such is shown by packet 500 in FIG. 5A, whose header contains the destination address of a broadcast mechanism 425, and contains the indication of the fault in field 502. This control packet 500 is sent

15   by SE-1 component 411B to broadcast mechanism 425 over link 441F. Broadcast mechanism 425 receives packet 500, and then creates and sends one or more packets to the I/O interfaces. An example of such is shown by packet 510 in FIG. 5B. Header field 511 contains the destination address of an I/O Interface 410, and field 512 contains the indication of the fault. One or more packets 510 are then sent over links 429 to each of

20   the I/O interfaces 410.

The processing by one embodiment of a broadcast mechanism is illustrated in FIG. 7A. Referring to FIG. 7A, processing begins at step 700, and proceeds to step 702 where a control packet containing a fault indication is received. Then, in step 704, one or more control packets are created and broadcast, multicast, and/or sent to each of the I/O

25   interfaces.

In one embodiment, a control packet containing a fault indication is sent to two or more different broadcast mechanisms to increase the likelihood of all I/O interfaces receiving the fault indication. As would be understood by one skilled in the art, these and

other variations are contemplated and accommodated by the extensive number of possible embodiments.

FIG. 7B illustrates one embodiment of the processing by an I/O interface for maintaining one or more data structures containing fault indications. Referring to FIG. 7B,

5   processing begins at step 720, and proceeds to step 722 where the I/O interface receives a packet containing a fault indication. Next, in step 724, the I/O interface updates one or more data structures it maintains of fault indications. These data structures may be updated to directly indicate the received indications of faults. In some embodiments, an additional thresholding function is performed to disable traffic from being sent to a

10   destination when the number of paths leading to the destination falls below some predetermined threshold value (e.g., number of paths, percentage of total paths, etc.). This predetermined threshold value may be preconfigured, or determined during operation to adjust to conditions of the packet switching system. For example, if nine of ten paths leading to a destination are unavailable, then traffic being sent to the destination may be

15   required to be sent over some smaller number of paths leading to the destination than when the packet switching system is fully operational. This may lead to congestion inside the packet switch system and/or congestion at the input interfaces of the packet switch system. By disabling the traffic to the destination from some or all of the input interfaces, this traffic congestion situation may be avoided. Processing then returns to step 722 to

20   receive more fault indications.

FIGs. 6A-B illustrate two of many different embodiments of data structures that could be maintained by an I/O interface to indicate fault conditions within the packet switching system. Data structure 600 is a table, typically implemented as a two-dimensional array or linked list, which maintains an entry for each output component

25   602 that is reachable over each interconnection network 601. In this manner, an I/O interface can easily determine which interconnection networks are available to deliver a particular packet destined for a particular output component (e.g., an I/O interface, line card, or port thereof, etc.) by looking at the appropriate column of data structure 600.

14

In one embodiment, data structure 610 is maintained in place of, or in addition to, data structure 600. One embodiment of data structure 610 is a link status vector 612, typically implemented as a bit vector, which indicates a link status for each interconnection network 611 to which it is coupled. In one embodiment, a bit is cleared in data structure

5   610 if the I/O interface is prevented from sending to the particular interconnection network 611.

FIG. 8 illustrates one embodiment's use of data structures 600 and 610 in determining a route for a particular packet. A set of possible routing paths is determined and maintained by a routing mechanism, which can be represented by bit vector 810. By

10   performing an AND function with output availability table 820 (e.g., the column of data structure 600 corresponding to the particular destination) and link status vector 830 (e.g., link status vector 612), a set of possible paths for the particular packet is generated and represented by bit vector 840. Bit vector 840 is then used by a routing mechanism to select one of the possible paths (e.g., corresponding to a set bit in bit vector 840), such as

15   by using a round-robin, random, even distribution, or other deterministic or non-deterministic method. In some embodiments, as previously described herein, a thresholding function is used to set the values corresponding to a particular output of the output availability table 820 to zero when the number of available paths represented in the routing paths data structure 810 falls below a predetermined threshold value.

20   FIG. 7C illustrates one embodiment of an I/O interface for determining a route for, and sending packets using the maintained one or more data structures of fault indications. Processing begins at step 740, and proceeds to step 742 where a particular packet is received to be sent to a particular destination. Next, in step 744, a routing mechanism determines a particular route for the received packet from those routes indicated as

25   available in the maintained one or more data structures. One embodiment of this processing was previously described in relation to FIG. 8.

Next, if there is a possible route for the received packet, as determined in step 746, then the packet is sent over the route determined in step 744. Otherwise, in step 750, the

15

packet may be dropped packet as the packet cannot be currently sent from the I/O interface to its intended destination. Additionally, some error handling (e.g., generating alarms, dropping the packet, sending control information to the packet's sender, etc.) may be performed in step 750. Processing then returns to step 742 to process more packets.

5   In view of the many possible embodiments to which the principles of our invention may be applied, it will be appreciated that the embodiments and aspects thereof described herein with respect to the drawings/figures are only illustrative and should not be taken as limiting the scope of the invention. To the contrary, the invention as described herein contemplates all such embodiments as may come within the scope of the following claims

10   and equivalents thereof.